

Wright State University

CORE Scholar

[Browse all Theses and Dissertations](#)

[Theses and Dissertations](#)

2010

Adaptive Thresholding for Detection of Radar Receiver Signals

Stephen R. Benson

Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all



Part of the [Electrical and Computer Engineering Commons](#)

Repository Citation

Benson, Stephen R., "Adaptive Thresholding for Detection of Radar Receiver Signals" (2010). *Browse all Theses and Dissertations*. 393.

https://corescholar.libraries.wright.edu/etd_all/393

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

Adaptive Thresholding for Detection of Radar Receiver Signals

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

By

Stephen R. Benson

B.S. COMPUTER ENGINEERING, Wright State University, 2008

2010

Wright State University

WRIGHT STATE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

September 19, 2010

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY Stephen Benson
ENTITLED Adaptive Thresholding for Detection
Of Radar Receiver Signals BE ACCEPTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF Master of Science in Engineering.

Chien-In Henry Chen, Ph.D.
Thesis Director

Kefu Xue, Ph.D.
Department Chair

Committee on Final Examination

Chien-In Henry Chen, Ph.D.

Marian Kazimierczuk, Ph.D.

Saiyu Ren, Ph.D.

Andrew T. Hsu, Ph.D.
Dean, School of Graduate Studies

ABSTRACT

Benson, Stephen Ray. M.S.Egr., Department of Electrical Engineering, Wright State University, 2010. *Adaptive Thresholding for Detection of Radar Receiver Signals*.

Digital microwave receivers play a critical role in many of today's modern radar tracking systems. The need for these digital receivers to push the boundaries in terms of bandwidth and input dynamic ranges (DR) is vital for their use in radar signal tracking. Significant research has been conducted in the area of the fast Fourier transform (FFT) to aid in continuing to enhance the performance capabilities of digital microwave receivers. However, with the advancement and increased complexity of these systems, the need for an efficient and effective adaptive thresholding technique is becoming ever more present.

The proposed adaptive thresholding technique utilizes signal magnitude evaluations and multi-stage signal scaling throughout a 128-point FFT in order to effectively determine the optimal threshold for the microwave receiver. The incorporation of a 10-bit dynamic kernel function, as well as 14-bit word size between FFT stages is used to aid in increasing receiver sensitivity, multi-tone instantaneous dynamic range (IDR) and spurious free dynamic range (SFDR) performance.

With the implementation of our adaptive thresholding technique, our receiver's maximum IDR is maintained between 34dB down to 24dB for input signal strengths ranging from -4dBm down to -32dBm. From simulation results incorporating the use of digitized data from our 10-bit Atmel ADC our Multi-Stage Scaling (MSS) receiver design is capable of obtaining an SFDR of 35.91dB using an input signal strength of -7dBm.

Contents

I.	INTRODUCTION AND PURPOSE	1
1.1	Digital Microwave Receivers.....	1
1.2	Fixed and Adaptive Thresholding	1
1.3	Motivation	2
1.4	Contribution.....	3
1.5	Document Organization	4
II.	FFT-Based Digital Microwave Receiver	5
2.1	Fast Fourier Transform(FFT)	10
2.2	Dynamic Kernel Function FFT	12
2.3	Digital Microwave Receiver Under Study	12
2.3.1	Past Work and Results	12
2.3.2	Receiver Design Utilizing a Single Initial Scaling Block	13
2.3.3	Improved Design Incorporating Multi-Stage Scaling Blocks	19
III.	Adaptive Thresholding	23
3.1	Adaptive Thresholding Technique Using Initial Stage Scaling Block.....	23
3.2	New Adaptive Thresholding Technique Utilizing Multi-Stage Scaling	26
3.3	Hardware Requirements of Both ISS and MSS Receiver Designs	30
IV.	Performance Evaluation	32
4.1	Performance Evaluation Based on Matlab Simulations Comparison of Adaptive Threshold Technique vs. Hard Set Threshold	32
4.2	Performance Evaluation of Xilinx System Generator Receiver Designs using System Generated Signals as Input.....	36
4.3	Performance Evaluation of Atmel Digitized Data for Multi-Stage Scaling and Single Stage Scaling Receiver Design.....	38
V.	Design Flow and Prototyping Hardware	40
5.1	Design Flow => Software – Matlab ->System Generator -> Xilinx ISE -> Chipscope.....	40
5.2	Prototyping Hardware – Delphi Board and Testbed Setup	44

VI. FFT-Based Digital Microwave Receiver	46
6.1 Contribution.....	46
6.2 Future Work	47

REFERENCES

List of Figures

Figure 2.1 Design flow for an 8-point decimation-in-frequency FFT	9
Figure 2.2 Radix-2 DIF butterfly operation	9
Figure 2.3 Unit circles that have been scaled by a factor of 8 and 2	11
Figure 2.4 Hierarchal view of the microwave receiver design	13
Figure 2.5 Design flow for Initial Scaling Stage scaling block	15
Figure 2.6 Design flow for Initial Scaling Block in ISS receiver design	16
Figure 2.7 Design flow for Initial Scaling Stage receiver design	18
Figure 2.8 Design flow for threshold setting in MSS receiver design	21
Figure 2.9 Design and data flow for MSS receiver	22
Figure 3.1 Hardware layout for ISS receiver's thresholding block	24
Figure 3.2 Example data flow throughout the ISS receiver's thresholding block	25
Figure 3.3 MSS thresholding hardware for secondary peaks	27
Figure 3.4 MSS thresholding hardware for primary peaks	28
Figure 3.5 Example data flow throughout the MSS receiver's thresholding block	29
Figure 4.1 Maximum obtainable IDR for 6-bit and 10-bit dynamic kernel receiver designs	34
Figure 4.2 Plot of the maximum obtainable two-tone IDR for primary signal strengths ranging from -4dBm down to -32dBm	36
Figure 4.3 Plot showing performance of ISS and MSS receivers using ideal inputs	38
Figure 4.4 SFDR performance of ISS and MSS receiver designs using digitized ADC data	39
Figure 5.1 Photo of the Virtex 4- SX55 FPGA board	41
Figure 5.2 Xilinx ISE synthesis hardware report for 64-point ISS receiver design	43
Figure 5.3 Top level view of a Delphi ADC3255 board	45

List of Tables

Table 3.1 Hardware requirements for ISS and MSS receiver designs.....30

List of Abbreviations

ADC	Analog to Digital Converter
ASIC	Application-Specific Integrated Circuit
CW	Continuous Waves
DSP	Digital Signal Processing
DFT	Discrete Fourier Transform
DR	Dynamic Range
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
FT	Fourier Transform
HDL	Hardware Description Language
IDR	Instantaneous Dynamic Range
ISS	Initial Scaling Stage
LSB	Least Significant Bit
MSB	Most Significant Bit
MSS	Multi-Stage Scaling
PMC	PCI Mezzanine Card
PW	Pulsed Waves
RAM	Random Access Memory
RF	Radio Frequency
ROC	Receiver-on-a-chip
SFDR	Spurious Free Dynamic Range
SNR	Signal-to-Noise Ratio
VHDL	Very-High-Speed Integrated Circuit Hardware Description Language
VTs	Variable Truncation Scheme
XSG	Xilinx System Generator

Acknowledgements

This thesis was conducted in conjunction with the NEWSTARS program, Wright-Patterson Air Force Base Research Laboratory, Dayton, Ohio.

First and foremost I would like to thank my advisor Dr. Henry Chen for his continuous guidance, mentoring, and unwavering support. Throughout this entire process he has given his support and expertise and challenged me to become a better researcher. I am extremely grateful for all that he has done for me; without him, none of this would have been made possible.

I would also like to take the time to thank my parents, for their constant encouragement. They have made countless sacrifices to ensure that I could further my education, and for that, I will be eternally grateful. Without their love and support, I would not be the person that I am today.

Furthermore I would like to recognize and thank Dr. Marian Kazimierczuk and Dr. Saiyu Ren for their willingness to serve on my thesis committee.

Lastly I would like to express my gratitude George Lee for his continuous mentoring and guidance. His knowledge and willingness to lend a helping hand has been priceless throughout this process.

Stephen Benson, MSEE 2010

*For my parents,
Ronald and Barbara Benson*

I. INTRODUCTION

1.1 Digital Microwave Receivers

With the continuous advancement of research and technological findings, digital microwave radar receivers are continuing to expand into the multi-gigahertz range as fine frequency resolutions become more precise and the need for more real-time data processing capabilities becomes more of a necessity.

Due to the nature of the environment that radar receivers operate in, a priori knowledge of the defining characteristics and total number of incoming signals is not present. This presents a high priority task of determining and setting an optimal threshold within the receiver to prevent the occurrence of false alarms, while trying to maintain a high receiver sensitivity and multi-tone instantaneous dynamic range (IDR) performance.

1.2 Fixed and Adaptive Thresholding for Signal Detection

Because clutter statistics can be highly unknown and variable, such as environmental noise, a fixed threshold may only be optimal for signals that are limited to a specific operational dynamic range. Because of this, the given receiver may be unable to detect weaker signals, due to the hindering constraint placed upon it by the fixed threshold.

An alternative approach may be to use a computationally intensive adaptive thresholding technique that utilizes the calculation of the mean and variance of the noise floor in the interested frequency spectrum in order to determine the optimal threshold

setting. These adaptive thresholding techniques may not always be realizable in systems with strict processing and memory constraints. Likewise, due to the timing constraints in many modern radar receivers, this approach may not be plausible for systems with real-time data requirements.

The proposed adaptive thresholding technique in this thesis is an efficient and effective method to increase overall receiver SFDR performance and signal detection rates while minimizing any increases in overall hardware usage. The adaptive thresholding technique described in this thesis evaluates the magnitudes of the incoming signal data supplied by our 10-bit Atmel Analog-to-Digital Converter (ADC), we are able to accurately determine an optimal threshold for the receiver in order to optimize detection rates and minimize false alarms.

1.3 Motivation

In previous years, the development and fabrication of devices capable of performing computationally intensive digital signal processing (DSP) algorithms on a single chip was monopolized by the use of application specific integrated circuits (ASIC). In general, however, ASICs tend to be a very costly venture, while building prototype units can be a lengthy process. In recent years, the capabilities of devices such as field-programmable gate arrays (FPGA) have created new possibilities for prototyping digital designs, including those involving DSP algorithms. With the emergence of FPGAs capability of processing large quantities of data and the added ability to implement complex systems such as fast Fourier transform (FFT)-based digital wideband

microwave receivers, their use has become highly favorable in the DSP world. FPGAs offer advantages such as easy reconfigurability, reduction of development time, and simpler testing and verification procedures. For these reasons, it is now possible to implement digital radar receivers on a single FPGA board, requiring a much smaller investment of time and monetary resources when compared to similar ASIC based designs.

For these reasons, a Xilinx Virtex 4 FPGA board coupled with an Atmel 10-bit ADC capable of sampling at 2.048GHz is the target platform for our FFT-based digital wideband microwave receiver. In terms of performance, the design of the FFT is the major contributor to the overall capabilities of the receiver. The ability of the FFT to accurately convert time-varying signals sampled in the time-domain into their frequency domain representation, consisting of their real and imaginary components is an important focus. By incorporating a series of techniques including an efficient 128-point FFT design, a 10-bit dynamic kernel function, multiple signal scaling blocks, an increase in the maximal word size between FFT stages, and an adaptive thresholding technique, we were able to considerably increase the overall receiver performance when compared to previous designs.

1.4 Contribution

Current simulation data shows promising performance results from our radar receiver design. Matlab simulations show a maximum obtainable two-tone IDR of 34dB while utilizing a primary input signal with strengths ranging from -4dBm down to -

15dBm. Performance evaluations of the receiver within Xilinx System Generator (XSG) show a maximum sensitivity of -45dBm when utilizing digitized data from our 10-bit Atmel ADC as input. SFDR performance remains at 9.62dB with an input signal strength of -45dBm using digitized data. The receiver is also capable of achieving a maximum SFDR of 35.91dB with an input strength of -7dBm when using digitized data from our Atmel ADC.

1.5 Document Organization

This thesis is comprised of seven chapters. Chapter I discusses background information for FFT-based digital microwave receivers and their implementation within FPGAs and various threshold setting methodologies. Chapter II provides a discussion of the fast Fourier transform and discrete Fourier transform algorithms as well as a hardware overview of previous and current receiver designs. Chapter III discusses the adaptive thresholding algorithm for both Initial-Stage Scaling (ISS) and Multi-Stage Scaling (MSS) receiver designs. Chapter IV covers performance evaluations from Matlab and Xilinx System Generator simulations for the receiver design under study while using a fixed threshold as well as our adaptive thresholding technique. The methodology for our design process and prototyping hardware is covered in Chapter V. Hardware usage statistics and FPGA verification results are discussed in Chapter VI. Finally, in Chapter VII is the conclusion and discussion of future work.

II. FFT-Based Digital Microwave Receiver

2.1 Fast Fourier Transform (FFT)

The fast Fourier transform is a highly used algorithm in the digital signal processing world to compute the discrete Fourier transform (DFT). The Fourier transform converts a finite set of samples taken in the time-domain into a series of samples represented within the frequency-domain. Historically, the direct computation of the DFT is not calculated within a design due to the high computational complexity. However, a series of efficient algorithms have been defined by Cooley and Tukey [14] to decrease the computational complexity by a substantial amount. The fast Fourier transform uses a divide and conquer method to reduce the computational complexity and required hardware for the computation of the DFT.

The fast Fourier transform computes the DFT for a given input data series $x(n)$ with a length N , and is defined as $X(k)$. Eq. (3.1) below shows the formula for computing the DFT,

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\left(\frac{2\pi}{N}\right)kn}, \quad 0 \leq k \leq N-1 \quad (3.1)$$

Typically, the more commonly used form for the fast Fourier Transform uses the generalized formula for the kernel function as defined in Eq. (3.2),

$$W_N^{nk} = e^{\frac{-j2\pi nk}{N}} \quad (3.2)$$

Substituting Eq. (3.2) into Eq. (3.1) allows Eq. (3.1) to be re-expressed as.

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad 0 \leq k \leq N-1 \quad (3.3)$$

The calculation of the kernel function, also known as the twiddle factor, is the major contributor of the computational complexity for calculating the DFT. To aid in reducing this complexity of calculating the twiddle factors, it is possible to expose the symmetric and periodic properties of the FFT algorithm. Euler's formula, as defined in Eq. (3.3) aids in exposing these properties,

$$e^{j\theta} = \cos(\theta) + j \sin(\theta) \quad (3.3)$$

Substituting our twiddle factor definition into Euler's formula, we get Eq. (3.4),

$$W_N^{nk} = e^{\frac{j2\pi nk}{N}} = \cos\left(\frac{2\pi nk}{N}\right) + j \sin\left(\frac{2\pi nk}{N}\right) \quad (3.4)$$

By substituting $k + \frac{N}{2}$ for k into our twiddle factor equation, we get the following,

$$W_N^{n(k+\frac{N}{2})} = e^{\frac{-j2\pi n(k+\frac{N}{2})}{N}} \quad (3.5)$$

$$= e^{\frac{-j2\pi nk}{N}} e^{-j\pi n} \quad (3.6)$$

Substituting part of Eq. (3.6) into rectangular form gives the formula,

$$e^{-j\pi n} = \cos(-\pi n) + j \sin(-\pi n) = -1 \quad (3.7)$$

From Eq. (3.6) and Eq. (3.7), the symmetric property of the DFT can be seen in Eq. (3.8),

$$W_N^{k+\frac{N}{2}} = -W_N^k \quad (3.8)$$

The symmetric property of the DFT is useful for reducing computational complexity, because it shows that half of the twiddle factors are able to be represented with their complex conjugate.

The periodic property of the DFT is proven in a similar manner. By substituting $k+N$ for k into Eq. (3.4),

$$\begin{aligned} W_N^{n(k+N)} &= e^{\frac{-j2\pi n(k+N)}{N}} \\ &= e^{\frac{-j2\pi nk}{N}} e^{-j2\pi n} \end{aligned} \quad (3.9)$$

Substituting part of Eq. (3.9) into rectangular form gives the formula,

$$e^{-j2\pi n} = \cos(-2\pi n) + j \sin(-2\pi n) = 1 \quad (3.10)$$

From Eq. (3.9) and Eq. (3.10), the symmetric property of the DFT can be seen in Eq. (3.11),

$$W_N^{k+N} = W_N^k \quad (3.11)$$

For the purpose of this research, a decimation-in-frequency FFT is used to convert the time-domain samples into their frequency-domain representations. The original Eq. (3.3) is broken up into equal parts, each representing $\frac{N}{2}$ points of the entire sequence of N .

The first part will represent the first $\frac{N}{2}$ points, while the second part will represent the second $\frac{N}{2}$ points. This breakup marks the beginning of the divide and conquer approach.

The described equation is defined below as Eq. (3.12),

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(n) (W_N^{nk}) + W_N^{\frac{Nk}{2}} \sum_{m=0}^{\frac{N}{2}-1} x\left(n + \frac{N}{2}\right) W_N^{nk} \quad (3.12)$$

With the general form, it is possible to expose the periodicity and symmetric properties of the DFT to decimate Eq. (3.12) into even and odd samples. The result of this operation is seen in the resulting equations listed below.

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} (x(n) + x(n + \frac{N}{2})) , \quad 0 \leq k \leq \frac{N}{2} - 1 \quad (3.13)$$

$$X(2k + 1) = \sum_{n=0}^{\frac{N}{2}-1} (x(n) - x(n + \frac{N}{2})) , \quad 0 \leq k \leq \frac{N}{2} - 1 \quad (3.14)$$

Equations (3.13) and (3.14) each represent an $\frac{N}{2}$ -point DFT, which can be further decimated for $\log_2(N)$ stages utilizing $\frac{N}{2}$ -radix-2 DFTs. By utilizing the periodicity and symmetric properties of the, it is possible to reduce the original required calculations from $O(N^2)$ complex additions and multiplications down to $O(N \log_2 N)$ complex additions and $O(\frac{N}{2} \log_2 N)$ complex multiplications. The decimation process and design flow can be seen on the following page for an 8-point decimation-in-frequency FFT in Fig. 2.1.

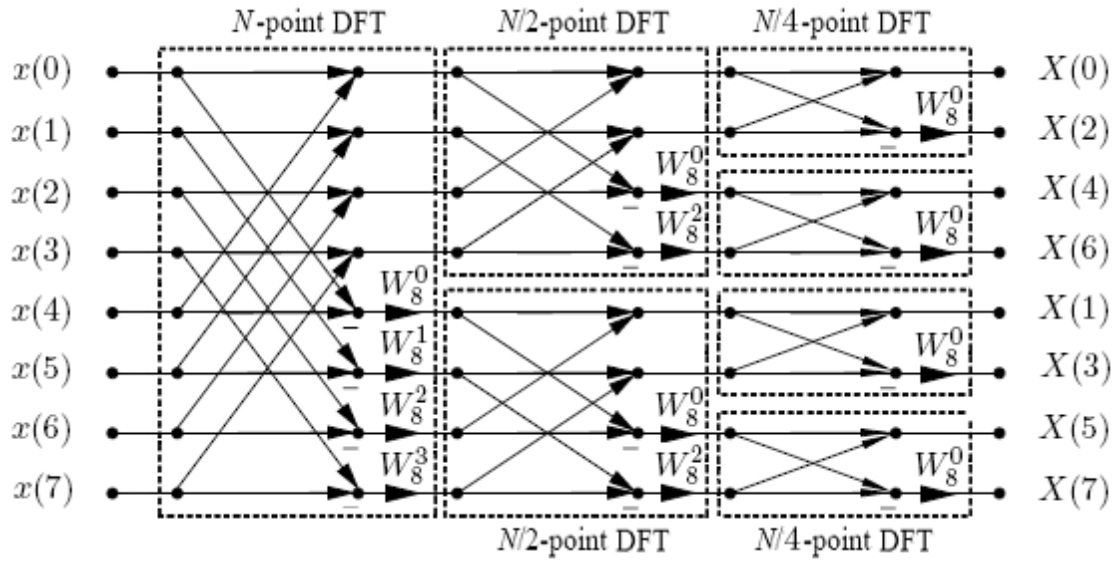


Figure 2.1 Design flow for an 8-point decimation-in-frequency FFT [17]

Fig. 2.2 shows the complex additions and multiplications required throughout the butterfly operations within each stage of our DFTs.

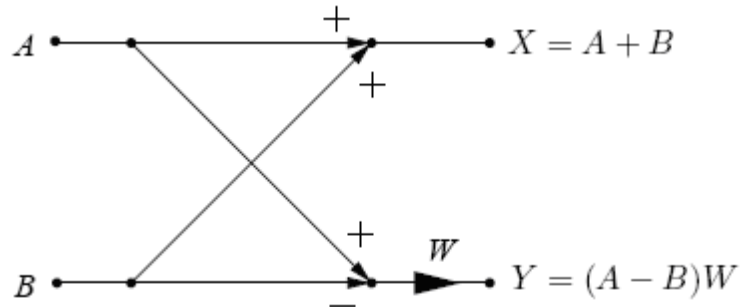


Figure 2.2 Radix-2 DIF butterfly operation [17]

2.2 Dynamic Kernel Function FFT

Since the target platform for our receiver design is an FPGA whose resources will always be limited in size, special consideration needed to be taken into account for the digital representation of the FFT's kernel function. Twiddle factors, in general, are difficult to implement in hardware because of their real and imaginary parts, most of which, are comprised of values less than one. An implementation involving floating point numbers can exponentially increase overall hardware usage, making it an unfavorable option for digital receiver design. A dynamic kernel function, however, provides an accurate estimation for the FFT's twiddle factors. This implementation also does not require the use of floating point numbers, thus minimizing the amount of hardware resources required.

For an N-point FFT design, a total of N kernel functions are required. These kernel functions can be represented within a unit circle, where each kernel function is composed of a real and imaginary part, and are equally spaced throughout the unit circle. These kernel functions need to be multiplied by the incoming data within the butterfly operations in the FFT. Since multipliers require large amounts of hardware resources to implement, a simple shift and add methodology is used to replace all multiplication operations within our receiver design.

Because all kernel functions defined within the unit circle are represented by a value of one or less, when all values within the unit circle are scaled up by a common factor, it is possible to represent the twiddle factors using a fraction that is composed of an integer numerator as well as an integer denominator. The number of bits required to represent the twiddle factors is dependent on the factor by which the unit circle was

scaled up by. In our receiver design, the unit circle is scaled up by a factor of 512, since we use a two's complement data representation, we require 10 bits to represent each twiddle factor. Fig. 2.3 below shows two unit circles, one that has been scaled up by a factor of eight, and another that's been scaled up by a factor of two.

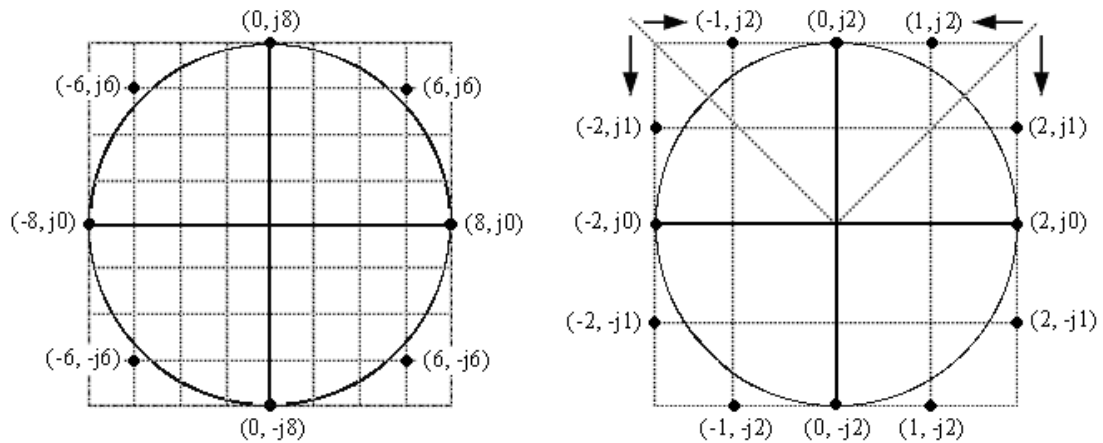


Figure 2.3 Unit circles that have been scaled by a factor of 8 and 2 [17]

The process by which the twiddle factors are created involve a series of shift and add operations to “multiply” the incoming data by the twiddle factor’s integer numerator. Because the unit circle is always scaled up by a factor of 2, the denominator simply requires the appropriate number of shifts right to complete the fractional “multiplication”. As the unit circle is scaled up by larger factors, it’s approximation to the ideal FFT’s twiddle factors become more accurate. The trade off for this improved accuracy is an increase in the number of bits and hardware to represent the twiddle factors.

2.3 Digital Microwave Receiver Under Study

This section will first cover previous receiver designs and performance evaluations. From there, the discussion will move on to cover our current receiver design flow as well as a discussion of the multiple variations of our microwave receiver that utilize different adaptive thresholding algorithms. For all current receiver designs, the use of a 128-point FFT with a 10-bit dynamic kernel function is incorporated. Also, the maximum word size permitted between FFT stages is maintained at 14-bits for current designs as well.

2.3.1 Past Work and Results

Previous research has paved the way for continuous advancement in our radar receiver designs. The original mono-bit design, which our current receiver design has evolved from, utilized a 2-bit ADC sampled at 2.5GHz. It was capable of achieving a two-tone instantaneous dynamic range of 5dB [16]. Later designs included a 2.5-GSPS digital receiver-on-a-chip (ROC). Significant improvements had been achieved with the ROC design, as it was capable of achieving a two-tone instantaneous dynamic range of 18dB [2]. A third design, also targeted for an ASIC platform was an extension of the 2.5-GSPS digital ROC design which incorporated the use of various windowing functions to improve the two-tone IDR performance of the receiver. With the use of an improved windowing function it was able to achieve a two-tone IDR of 23dB [16]. The threshold setting scheme for each of these designs was based on a fixed threshold implementation.

A more recent radar receiver design was implemented with a target platform of an FPGA and used a semi-adaptive thresholding scheme. This design incorporated a dual-

thresholding scheme, for which the threshold was determined based on the strength of the incoming signals. The use of a variable truncation scheme allowed for the precise selection of the optimal 8-bits to keep from the 10-bit ADC data supplied. This receiver design was capable of achieving a two-tone IDR of 18dB [8].

2.3.2 Receiver Design Utilizing a Single Initial Scaling Block

Our original adaptive thresholding microwave receiver design is known as the Initial-Stage Scaling (ISS) design. The ISS receiver incorporates the use of a 128-point FFT with a 10-bit dynamic kernel. It also utilizes a maximum word size of 14-bits between FFT stages. Fig. 2.4 below shows a hierarchal overview of the microwave receiver design.

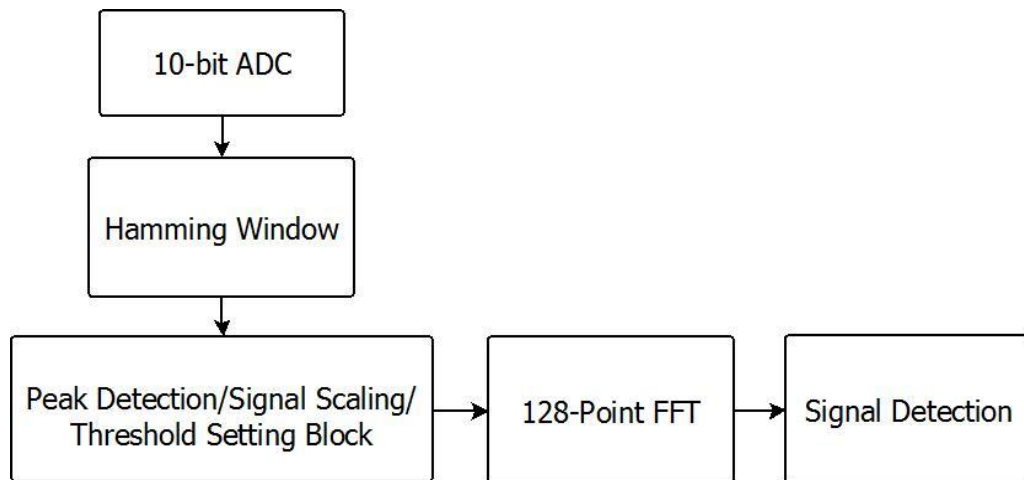


Figure 2.4 Hierarchal view of the microwave receiver design

The current receiver design only places a limitation on the maximum word size for data allowed to pass between FFT stages. To accomplish this, it is assumed that data traversing through the FFT butterfly blocks will only grow by a maximum of one bit. To

negate this effect, and essentially limit the maximum word size to 14-bits, as data leaves an FFT stage it is automatically truncated by one bit. This ensures that the data within our FFT will not exceed 14-bits.

Data is transferred from our 10-bit Atmel ADC into a series of demultiplexers to provide a set of 128 parallel inputs for the FFT. Before the data is sent to the first stage of the FFT, it is first passed into a scaling block. This scaling block is used to scale weaker signals up, to allow for greater overall receiver sensitivity as well as increasing the SFDR performance of the receiver.

The scaling block is comprised of series of comparators, to first determine the peak magnitude within the incoming signals. The peak is then compared with a set of scaling cutoffs to determine the proper factor by which to scale the signals up. These scaling cutoffs are based on powers of 2. The cutoffs directly represent the number of bits required to represent the given magnitude values of the incoming signals. Since a two's complement number system is used, a 10-bit representation is required for any magnitude value greater than 255. Similarly, a magnitude value between 128 and 255 requires 9-bits to accurately represent it, this pattern continues on for lower magnitude values and bit representations. The purpose of choosing this scaling scheme is to minimize the hardware needed to scale incoming signals. Because all scaling is based on powers of two, any multiplication or scaling up of signals can be replaced by a simple shift left operation, or padding a series of zeroes to the least significant bit (LSB) of the number. A design flow can be seen on the next page in Fig. 2.5 for the initial scaling block.

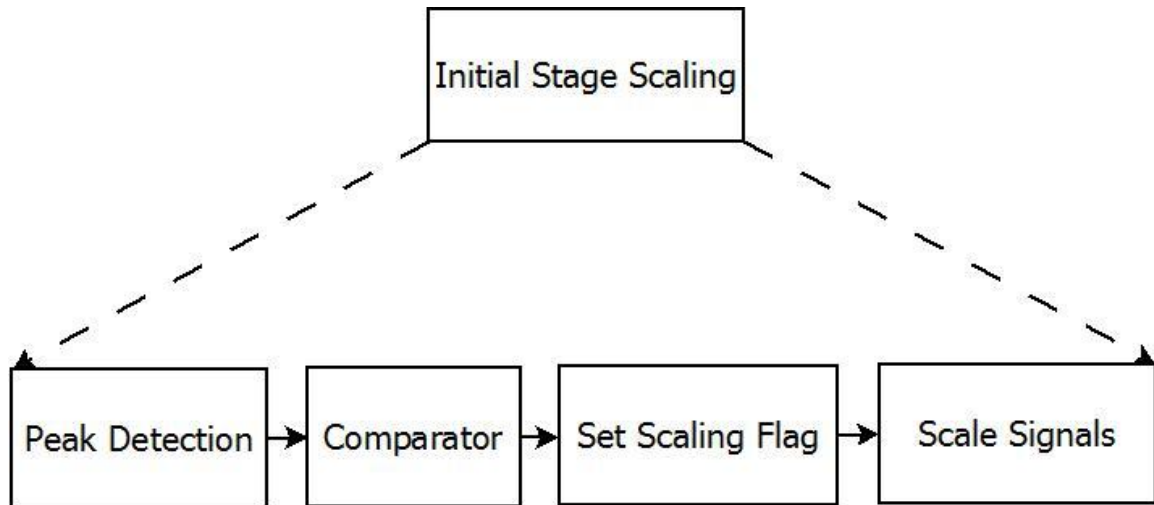


Figure 2.5 Design flow for Initial Scaling Stage scaling block

Once the magnitudes of the incoming signals have been evaluated, and the appropriate scaling factor has been chosen, it is at this point that the scaling flag is set. The scaling flag is a numerical representation for the factor by which the signals are scaled up. In the initial scaling stage, a scaling flag of '0' represents that the signals were scaled up by a factor of 16, meaning the incoming data was scaled from its original 10-bit representation from the ADC up to 14-bits. Similarly, a scaling flag of '1' signifies the incoming signals were scaled up by a factor of 32. The scaling factors are always based upon powers of two; the maximum that any signal can be scaled up by is 512, which uses a scaling flag of '5'. A flow chart depicting the operation of the initial scaling block can be seen on the next page in Fig. 2.6.

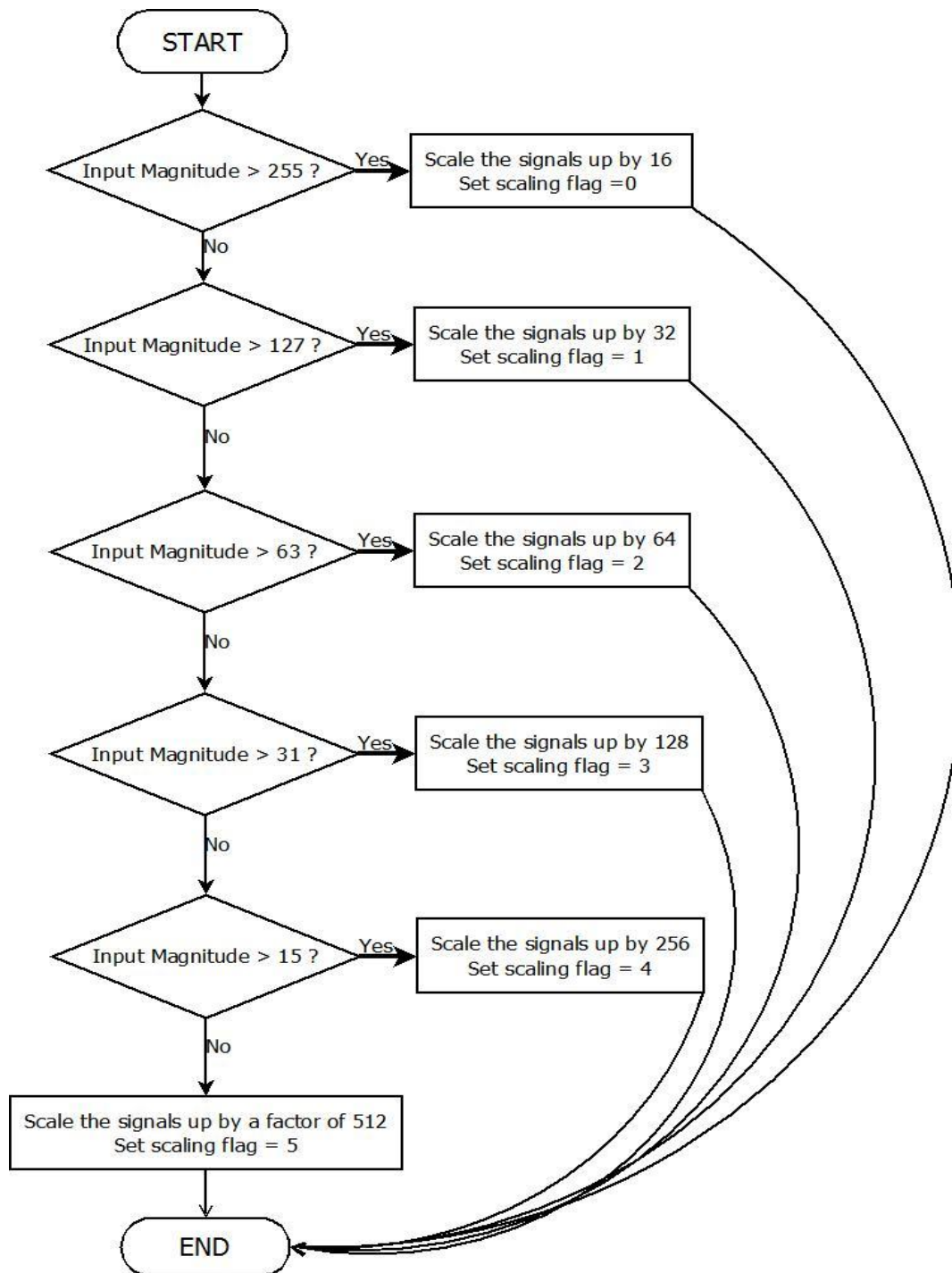


Figure 2.6 Design flow for initial scaling block in ISS receiver design

The scaling flag set by the initial scaling stage is the main determinant for which threshold to use within the receiver. After the scaling flag is set by the initial scaling stage, it is sent to the thresholding block, located at the end of our receiver design. As the data leaves the final stage of the FFT, it first goes through an $R + 1/2 I$ approximation if $R > I$ (or $I + 1/2 R$ if $R < I$) for an $R^2 + I^2$ magnitude evaluation. After leaving the $R + 1/2 I$ magnitude evaluation block, the data is passed into a peak detection block. This block finds the highest primary and secondary local peaks from the current FFT data. The frequency bins of the primary and secondary peaks are also gathered with the magnitude values of the peaks and are delivered to the thresholding block. The purpose of the thresholding block is to compare the currently set threshold with the magnitude values of the current peaks. If the magnitudes of the peaks are greater than the threshold, they are outputted from the thresholding block, otherwise a '0' is outputted for both the magnitude and frequency. A hierarchical view of the data flow of our ISS receiver design is shown on the next page in Fig. 2.7.

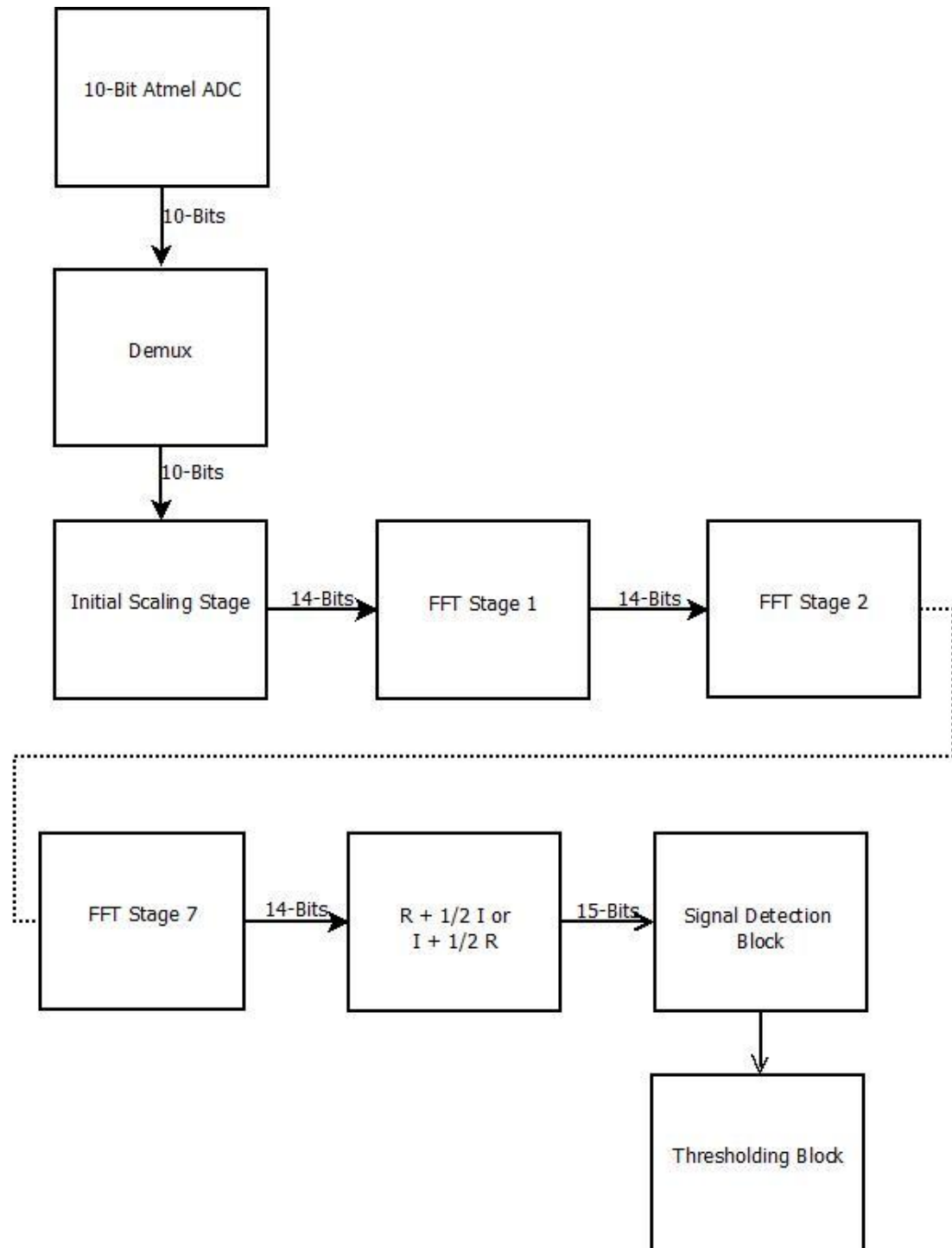


Figure 2.7 Design flow for Initial Scaling Stage receiver design

2.3.3 Improved Design Incorporating Multi-Stage Scaling Blocks

The Multi-Stage Scaling (MSS) receiver design was based upon our original ISS receiver design, with additional hardware added to increase the precision of our adaptive thresholding algorithm. The major of differences between the ISS and MSS receiver are in their implementation and use of the various scaling stages as well as variations in the thresholding algorithm implementation. As the name implies, the MSS receiver design uses numerous scaling stages placed between FFT stages throughout the receiver design. The use of multiple scaling stages helps to increase receiver detection rates and improve receiver SFDR performance. Similar to the ISS receiver design, the MSS receiver design still incorporates the use of the initial scaling stage found in the ISS receiver. The initial scaling flag is also used to aid in determining the appropriate threshold for the receiver. Data from the 10-bit Atmel ADC is still scaled from its original 10-bit value up to 14-bits. However, after FFT stage one, additional scaling blocks are implemented between every stage of the FFT up to FFT stage six. The goal of each of the scaling blocks is to maintain the signals at nearly their full scale values, while still avoiding any data saturation that could occur from over scaling. This implementation is useful due to the nature of data loss between FFT stages. In an ideal FFT, data passing through each stage would be allowed to grow by one bit every stage. In both ISS and MSS designs, however, one bit is always truncated after each stage to preserve data at or below 14-bits.

To accomplish our goals of maximizing receiver performance and reducing data saturation, scaling stages two through five all maintain or scale signals up to their 13-bit representation. The sixth and final scaling stage, located before FFT stage six, plays two

special roles in the MSS receiver design. First, it scales the signals up to their full 14-bit representations. Secondly, it is used similarly to the initial scaling stage of both ISS and MSS designs; however, scaling stage six sets a second scaling flag which is also used in the thresholding block to aid in threshold optimization. A flow chart depicting the use of both the initial scaling stage and scaling stage six for setting the threshold can be seen in Fig. 2.8 on the next page.

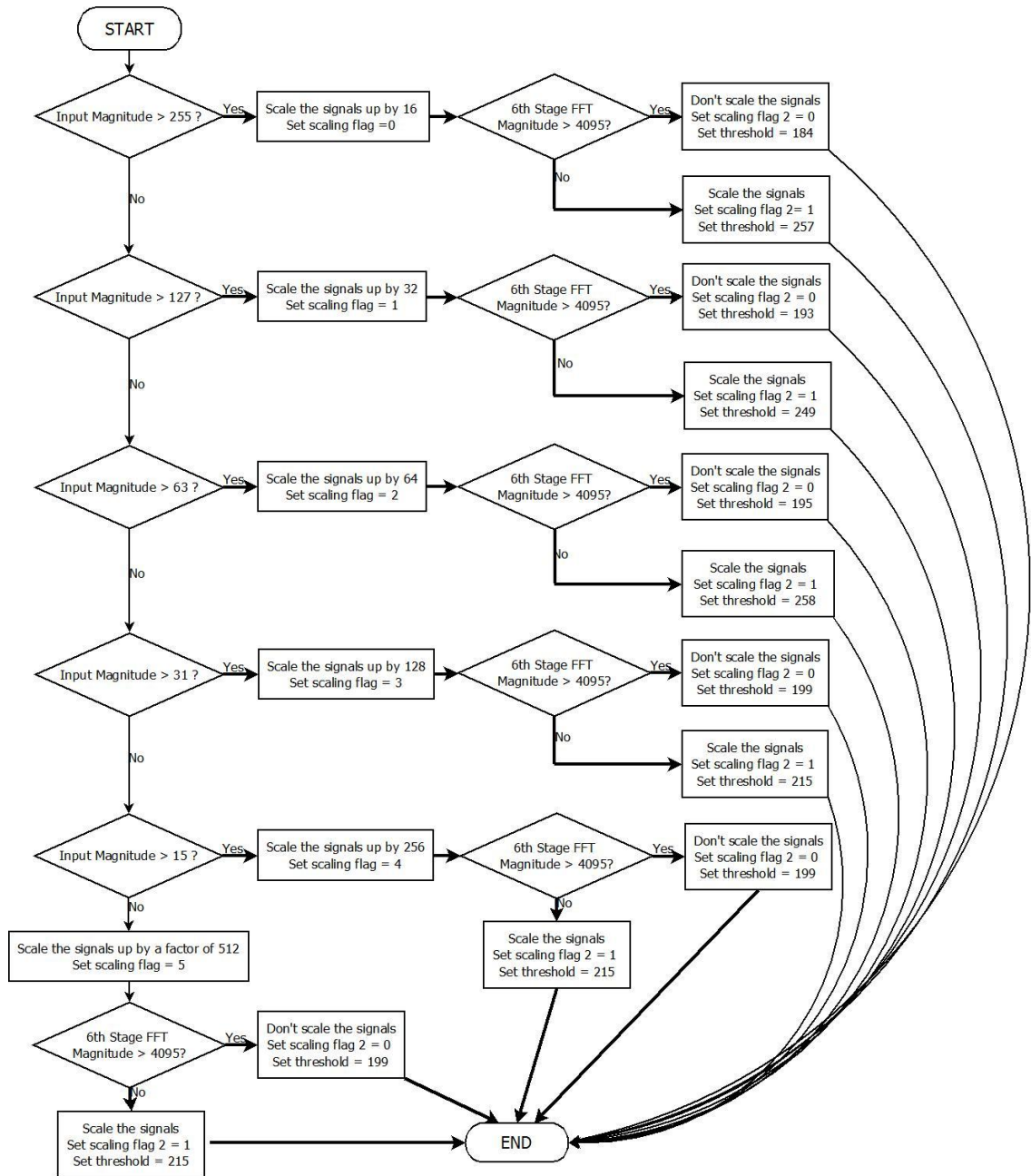


Figure 2.8 Design flow for threshold setting in MSS receiver design

Similar to the ISS design, after the data has passed through the seventh stage FFT, it passes through an $R + \frac{1}{2} I$ or $I + \frac{1}{2} R$ approximation of an $R^2 + I^2$ magnitude evaluation. It then travels through a peak detection block, which determines the primary and secondary peaks. The peaks and their frequency bins are finally passed through to the thresholding block to determine if the peaks should be considered as signals. A design and data flow chart can be seen below in Fig. 2.9

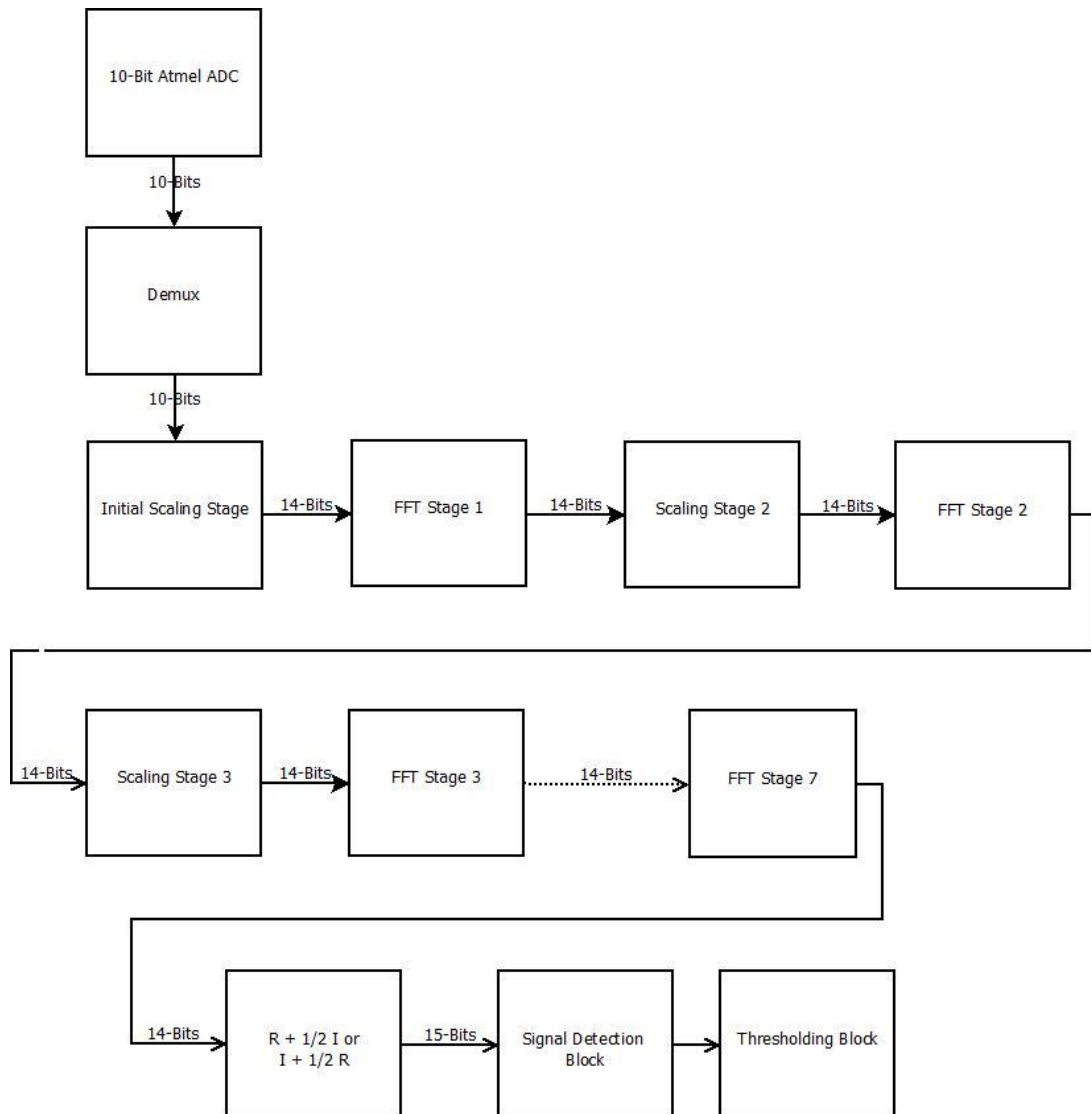


Figure 2.9 Design and data flow for MSS receiver

III. Adaptive Thresholding

3.1 Adaptive Thresholding Technique Using Initial Stage Scaling Block

For the ISS receiver design, the threshold is chosen solely based on the scaling flag set from the initial scaling stage. The actual thresholding block is comprised of two 6:1 multiplexers, two 2:1 multiplexers and a set of comparators. The inputs for the 6:1 multiplexers are pre-determined threshold levels. The scaling flag is used as the input for the select lines of both multiplexers. The primary multiplexer is used to pass the appropriate threshold when considering if a peak is a primary signal or not. Similarly the secondary multiplexer is used to pass the appropriate thresholds for determining if a signal should be considered as a secondary signal. Generally, the threshold values for the primary signal are larger than those for a secondary signal.

After the appropriate threshold has been selected from the multiplexers, its value is sent to a comparator, one of which compares the current primary peak with the threshold from the primary multiplexer, and another comparator that compares the current secondary peak with the threshold from the secondary multiplexer. If the current primary or secondary peak is larger than the predetermined threshold, the magnitude and frequency bin data is allowed to pass out of the thresholding block. The hardware makeup of the ISS receiver design thresholding block can be seen on the next page in Fig 3.1.

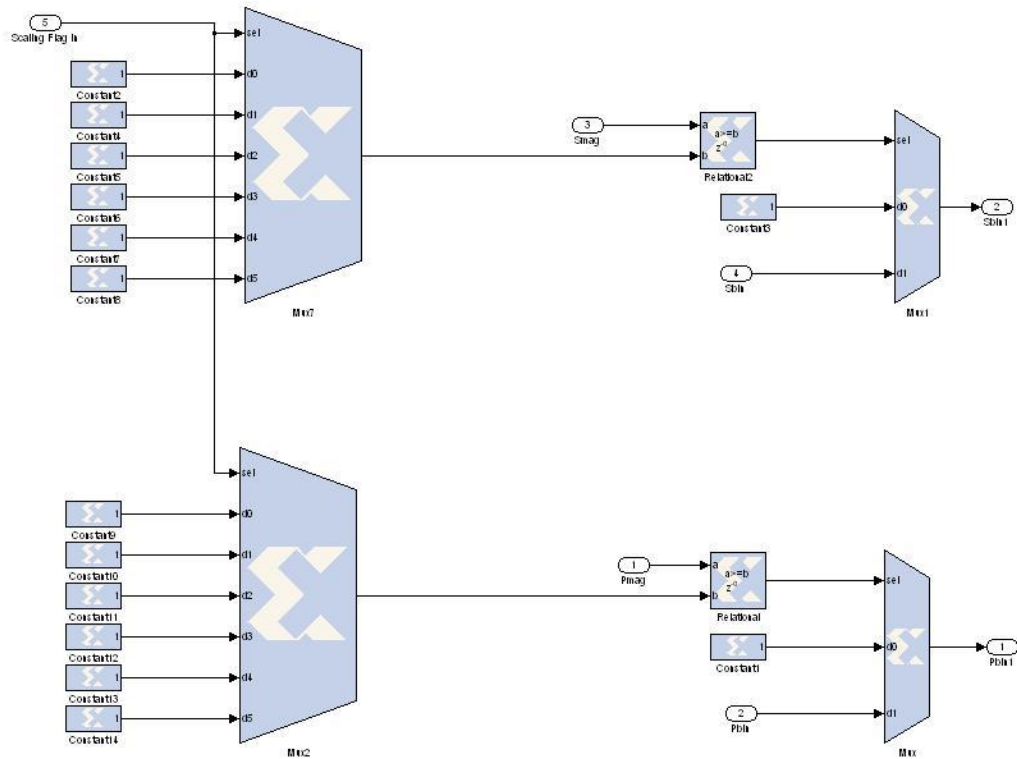


Figure 3.1 Hardware layout for ISS receiver's thresholding block

To further show the functionality of the thresholding block, Fig. 3.2 on the following page uses example magnitude values, thresholding values, and a defined scaling flag to show the data flow through the thresholding block. For this example, the primary scaling flag is set to a value of '2', thus selecting the defined thresholding constants for both 6:1 multiplexers as shown in the figure. The selected threshold data is passed through the 6:1 multiplexers and sent to their corresponding comparators. Following the data sent from the first multiplexer shows that the data is passed to a comparator, which evaluates the current secondary local peak, labeled as input "Smag", with the currently selected secondary threshold. If the current secondary local peak is greater than the threshold, the

frequency bin data for the secondary local peak, labeled as “Sbin”, is allowed to pass through the 2:1 multiplexer. If the current secondary local peak is not greater than the currently selected threshold, a ‘0’ is passed through the 2:1 multiplexer as the frequency bin data, denoting a signal was not detected. An identical procedure is used to determine and evaluate the primary peak and threshold.

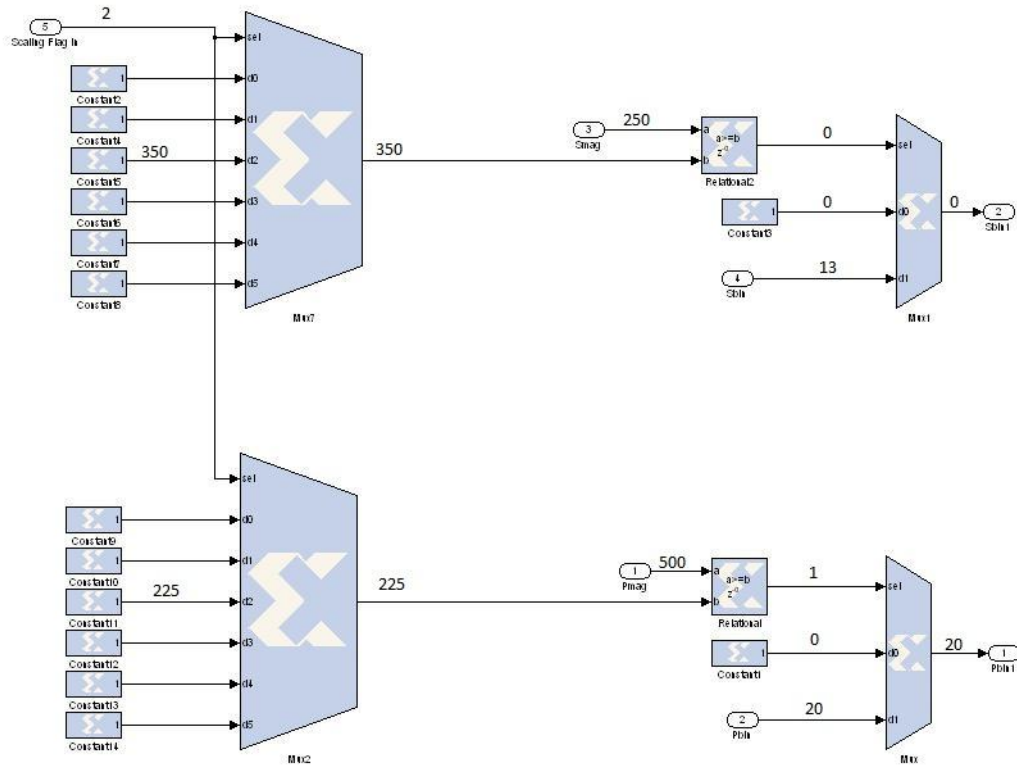


Figure 3.2 Example data flow throughout the ISS receiver's thresholding block

3.2 New Adaptive Thresholding Technique Utilizing Multi-Stage

Scaling

The MSS receiver design benefits from using information from multiple scaling stages in order to more accurately optimize the receiver threshold. As with the ISS receiver design, the initial scaling stage scales all incoming signals up to their 14-bit representation and sets a value for the scaling flag between '0' and '5', depending on the factor by which the signals are scaled up. However, when determining the final threshold, a secondary scaling flag is set by scaling stage six. Both scaling flags must be passed to the thresholding block before a threshold is set for the receiver. A slight increase in hardware is required to use both scaling flags to set the threshold. Where the ISS receiver design incorporated two 6:1 multiplexers, our MSS design requires eight 6:1 multiplexers, two 4:1 multiplexers, two 2:1 multiplexers, and two comparators.

Four different sets of thresholds are used for both the primary signal and the secondary signal. The concept for selecting the threshold is very similar to that used by the ISS design, except there is an additional level of thresholds to choose from. The primary scaling flag is used to select the appropriate threshold from any one of the 6:1 multiplexers. Each 6:1 multiplexer contains an individual set of thresholds based upon the factor by which the signals are scaled up in the sixth scaling stage.

The set of thresholds from which to use is determined by the secondary scaling flag. Both primary signal and secondary signal thresholds are set in this manner. Once the appropriate threshold has been determined, the primary and secondary peaks are compared to the threshold, only peaks which are greater than the threshold will be sent

out from the thresholding block. The following two figures, Fig. 3.3 and Fig. 3.4 depict the hardware within the MSS thresholding block used to set the thresholds for the primary and secondary signals.

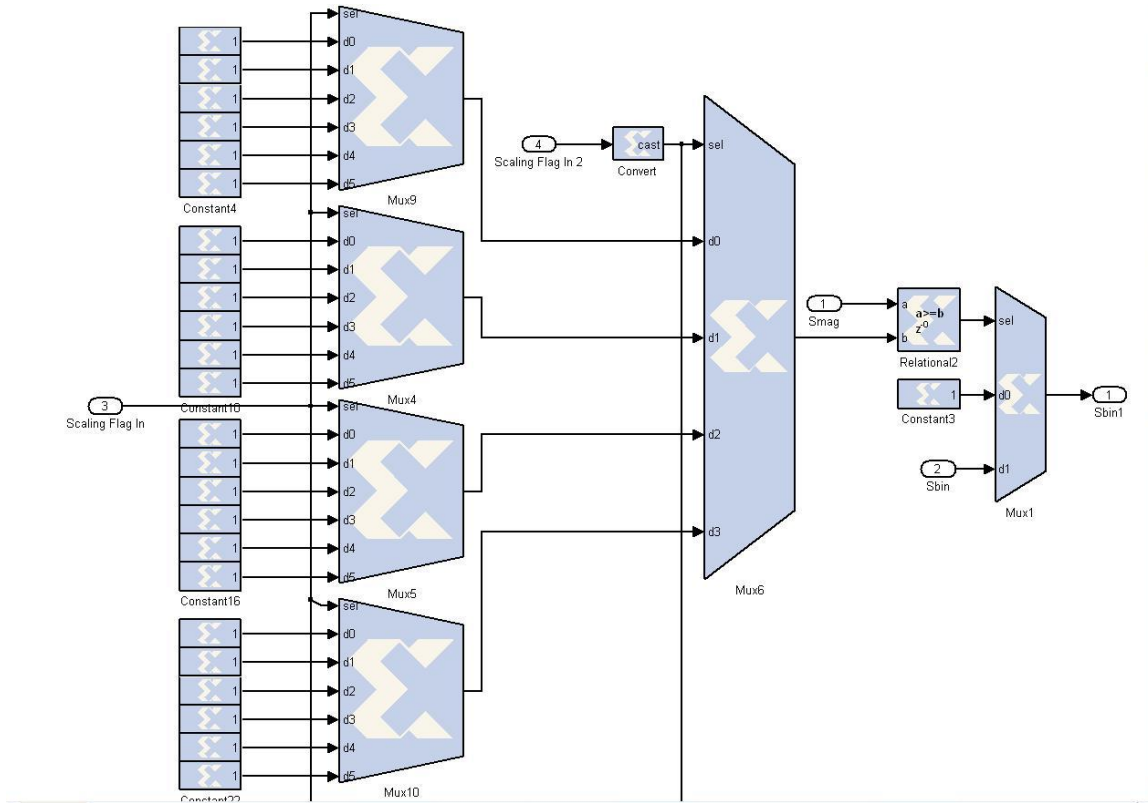


Figure 3.3 MSS thresholding hardware for secondary peaks

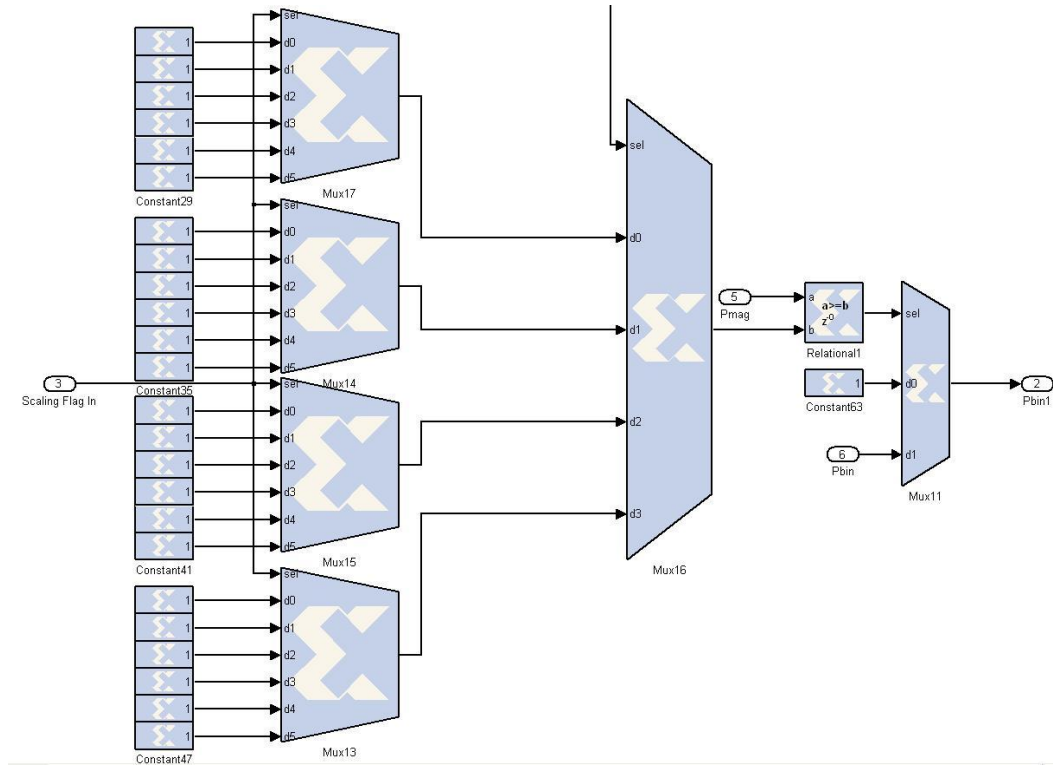


Figure 3.4 MSS thresholding hardware for primary peaks

The functionality of the MSS thresholding block can be more easily seen with the use of Fig. 3.5, placed on the next page, which shows the data flow for an example scenario. For this example, the primary scaling flag has a value of ‘5’, and therefore the fifth input for each 6:1 multiplexer is selected. To determine which of the four threshold values is to be used, scaling flag 2 is used the input select line for the 4:1 multiplexer shown in the figure. In this case, scaling flag 2 happens to be set to a value of ‘1’, which selects the input ‘200’, provided by the second 6:1 multiplexer. ‘200’ is now the currently set threshold for the receiver, and this value is passed onto a comparator, which compares the current secondary local peak input, labeled as “Smag”, with the currently set threshold. If the secondary local peak is greater than the threshold, the final 2:1 multiplexor will set the frequency bin of the secondary peak, labeled as “Sbin”, as its output, otherwise it will output a ‘0’ for the frequency bin, denoting no signal was

detected. An identical procedure is used to determine and evaluate the primary peak and threshold.

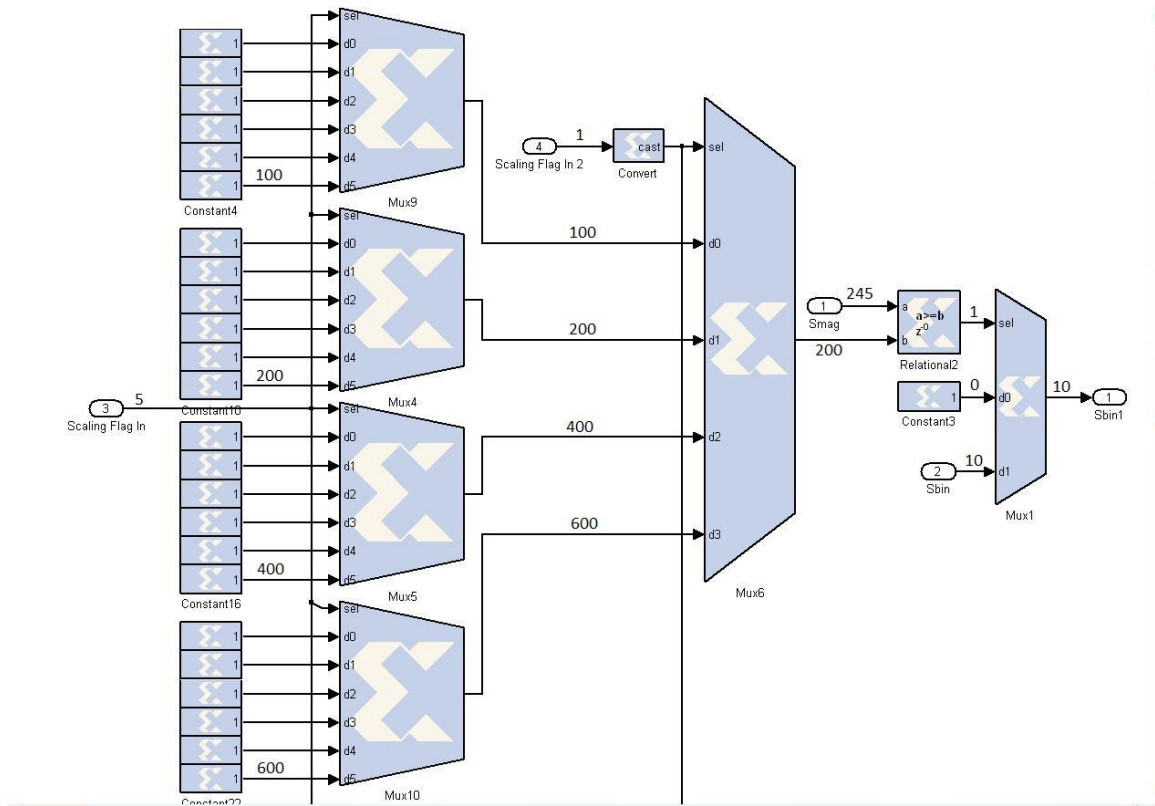


Figure 3.5 Example data flow throughout the MSS receiver's thresholding block

3.3 Hardware requirements of both ISS and MSS receiver designs

Hardware requirements were measured and recorded for both ISS and MSS receiver design in order to effectively measure their increase in performance vs. increase in hardware usage trade-offs. Through calculations, it was found that the majority of hardware usage was in direct relation to the FFT kernel functions and butterfly operations. The increased hardware due to our adaptive thresholding technique for both ISS and MSS designs was minimal when compared to the FFT hardware requirements of the receiver. Table 3.1 below shows the hardware requirements for both ISS and MSS receiver designs.

Hardware Usage Comparison			
	ISS	MSS	Overall Increase
1-to-128 TDD	1	1	0
Comparators	455	1253	798
14-bit Adder	1280	1280	0
16-bit Adder	3328	3328	0
14-bit Subtractor	1152	1152	0
16-bit Subtractor	160	160	0
Invertors	69	99	30
2-input AND Gates	5	35	30
6-input OR Gates	1	7	6
2-1 Mutliplexors	136	173	37
5-1 Mutliplexors	1	2	1
6-1 Multiplexors	2	26	24

Table 3.1 Hardware requirements for ISS and MSS receiver designs

As can be seen from the table above, both MSS and ISS designs require mainly an increase in the overall number of comparators within the receiver design. Comparators

are used to determine the strength of the incoming signals in order to accurately determine the appropriate scaling factors. The chart also shows the majority of hardware requirements for both receiver designs are due to the total number of adders and subtractors, which make up the FFT butterfly and kernel functions. It can also be seen that beyond the increase in comparators the our MSS design, there is only a small increase in basic logic gates and multiplexors.

In comparison to the overall hardware requirements of the receiver, both ISS and MSS receiver designs provide good performance increases vs. hardware requirements, making them a efficient and effective methodology for increasing receiver performance.

IV. Performance Evaluations

4.1 Performance Evaluation Based on Matlab Simulations Comparison of Adaptive Threshold Technique vs. Hard Set Threshold

Original implementations for the receiver design were based in software and created with the use of Matlab. The Matlab receiver design incorporated the use of our 128-point FFT with a 10-bit dynamic kernel. Simulations were also run using a 6-bit dynamic kernel to plot the performance differences between the two designs. The Matlab receiver design only incorporated a single scaling stage, thus its functionality was nearly identical to our ISS receiver design.

Simulations were run to test both the receiver's sensitivity and two-tone instantaneous dynamic range (IDR) performance. For all simulations, signal detection rates and false alarm rates were charted to monitor receiver performance. As a good performance metric evaluation, 10,000 simulations were run for all test bed setups. The first requirement to determine if a signal is considered detectable by the receiver is if the false alarm rate was 0 out of 10,000 simulations, or maintained a false alarm rate of less than 0.01%. The second requirement was that the signal must be detectable for a minimum of 80% of the 10,000 simulations.

To test the performance improvements of a 10-bit dynamic kernel vs. a 6-bit dynamic kernel, identical simulation setups were used for both designs. Two continuous wave (CW) signals with random frequencies were used as inputs for both receiver designs. The only stipulation placed upon the frequencies of the two signals was that they

must be maintained a minimum of three frequency bins apart. For our implementation, using a sampling frequency of 2.048GHz, three frequency bins of separation is equivalent to being separated by 48MHz.

A sweeping methodology was used to find the maximum IDR for all primary input signal strengths. Primary signal strengths were swept from -4dBm down to -18dBm, while secondary input signal strengths were ranged from 16 down to 36dB below the current primary signal. Our receiver design incorporating a 6-bit dynamic kernel achieved a maximum IDR of 34dB with a primary signal strength of -4dBm. However, for primary signal strengths ranging from -5dBm down to -18dBm, the design was only capable of achieving an IDR of 32dB.

Performance metrics for the receiver design utilizing a 10-bit dynamic kernel were measured in the same way. The receiver design utilizing a 10-bit dynamic kernel was able to match or outperform the two-tone IDR performance of our 6-bit dynamic kernel design for all primary signal strength varying from -4dBm down to -18dBm. A plot showing the maximum obtainable two-tone IDR values for both 6-bit dynamic kernel and 10-bit dynamic kernel receiver designs is shown on the next page in Fig. 4.1.

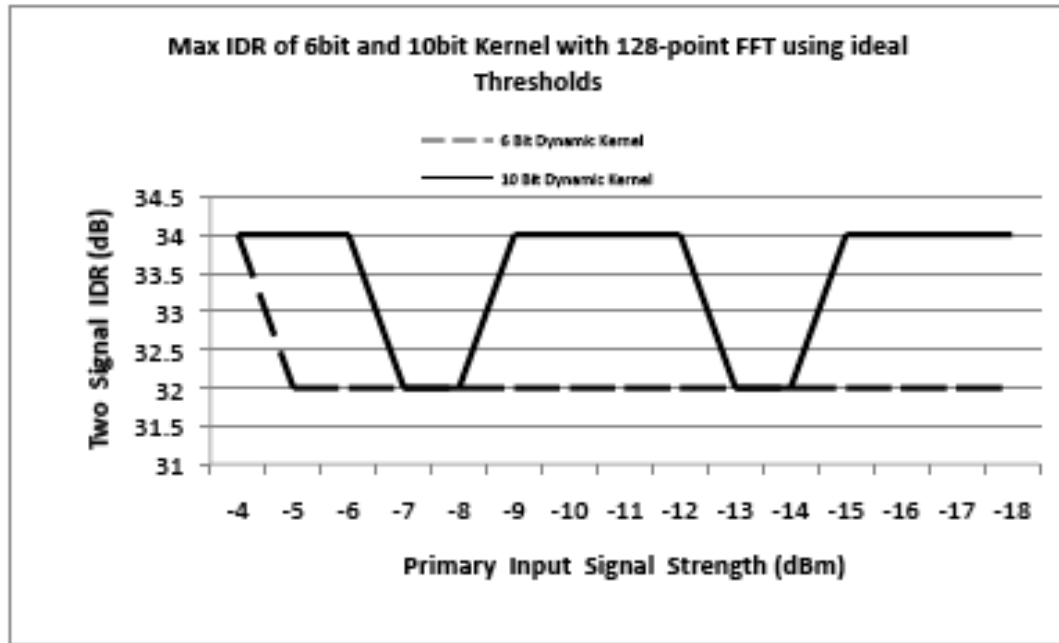


Figure 4.1 Maximum obtainable IDR for 6-bit and 10-bit dynamic kernel receiver designs.

To test the receiver's overall sensitivity, a CW signal was used as an input to the receiver. The frequency of this signal was generated randomly, and its magnitude was ranged from -4dBm down to -60dBm and decremented in 1dBm steps. Various thresholds were chosen to find the optimal threshold for each signal strength. 10,000 simulations were ran for each signal magnitude and threshold set. The receiver design was capable of detecting a single signal with an input strength of -53dBm with over 80% detection rate and less than 0.01% false alarm rate.

To test the receiver's two-tone IDR performance, a similar test setup was used. Both inputs for the receiver were CW signals whose frequencies were randomly generated. Similar to previous tests, the two signals were separated by a minimum of 48MHz. A

sweeping methodology was used to chart the performance of the receiver for varying primary and secondary input strengths. Input signal strengths for the primary signal ranged from -4dBm down to -32dBm and were decremented in 1dBm steps. Input signal strengths for secondary signals were swept from -16 down to -36dB below the currently set primary signal strength and were decremented in 1dB steps. Similar to the simulations ran to test the receiver's sensitivity, varying thresholds were tested for each set of primary and secondary signal strengths. 10,000 simulations were run for each set of primary signal strengths, secondary signal strengths and threshold settings. Detection requirements were maintained at a minimum 80% detection rate and less than a 0.01% false alarm rate. The receiver was able to maintain a two-tone IDR ranging between 32dB and 34dB for primary signal strengths between -4dBm and -22dBm. The receiver was capable of obtaining a two-tone IDR of 24dB for a primary signal strength of -32dBm. Fig. 4.2 on the next page plots the maximum obtainable two-tone IDR for primary signal strengths ranging from -4dBm down to -32dBm.

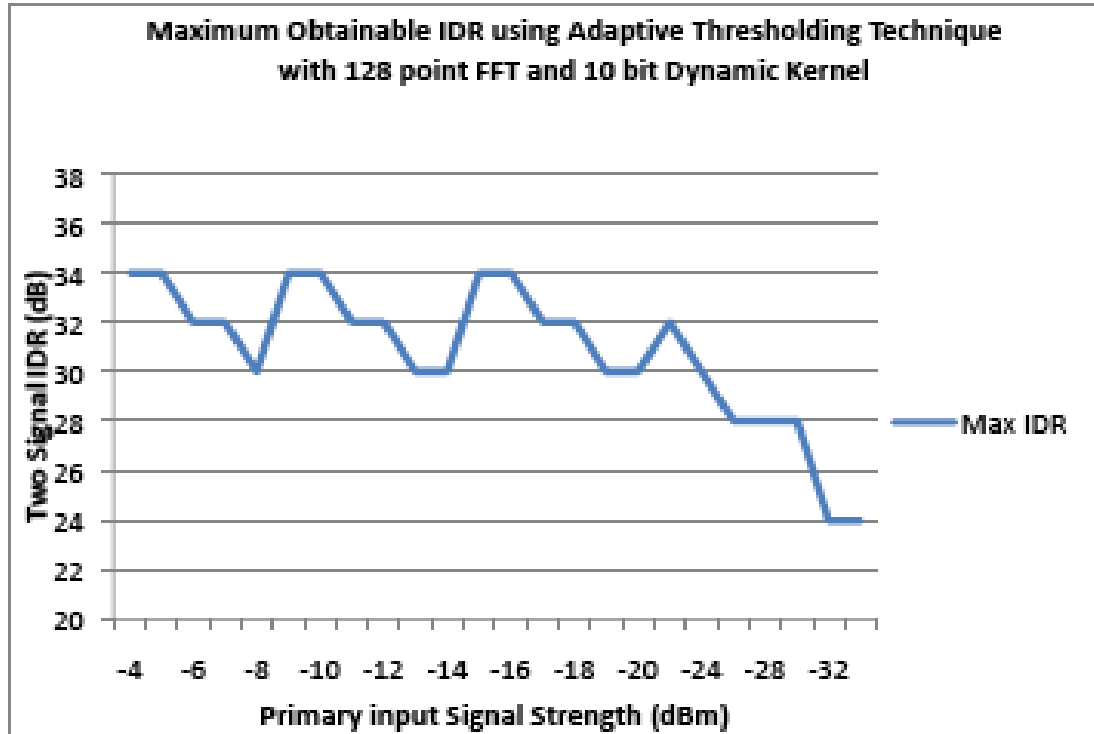


Figure 4.2 Plot of the maximum obtainable two-tone IDR for primary signal strengths ranging from -4dBm down to -32dBm.

4.2 Performance Evaluation of Xilinx System Generator Receiver

Designs using System Generated Signals as Input

After completing and evaluating all Matlab based simulations, it was necessary to implement and test our receiver design using Xilinx's System Generator (XSG) tools. Both ISS and MSS receiver designs were implemented within Xilinx System Generator. The first series of tests on the receiver designs were to determine the receivers' overall sensitivity as well as its maximum spurious free dynamic range (SFDR) when using ideal input signals. For these simulations, it was unnecessary to run 10,000 simulations due to the input data being perfectly ideal and non-fluctuating. Our test bed setup involved ranging a CW input signal from -1dBm down to -54.18dBm for both ISS and MSS

receiver designs. The reasoning for choosing an odd numbered minimal signal strength is simply because -54.18dBm represents the weakest signal representable as an input for our system based on our 10-bit ADC.

From our findings through simulations, it was found that when using an ideal signal as input, our ISS receiver design was capable of a maximum SFDR of 55.73dB with an input signal strength of -1dBm. The MSS receiver design, however, was capable of achieving an SFDR of 61.27dB using an input signal strength of either -1dBm or -2dBm. Both ISS and MSS receivers were capable of detecting a weak single signal input of -54.18dBm. Our MSS receiver design was able to match or outperform the ISS receiver design for all tested signal strengths except signals with an input strength of -40dBm and -50dBm. The ISS design outperformed our MSS design by obtaining a .04dB higher SFDR when using a signal strength of -40dBm by. The SFDR performance was much more dramatic with a signal strength of -50dBm, as the ISS achieved a higher SFDR of 4.55dB over the MSS receiver design. A chart plotting the performance of our ISS and MSS receiver designs can be seen on the following page in Fig. 4.3.

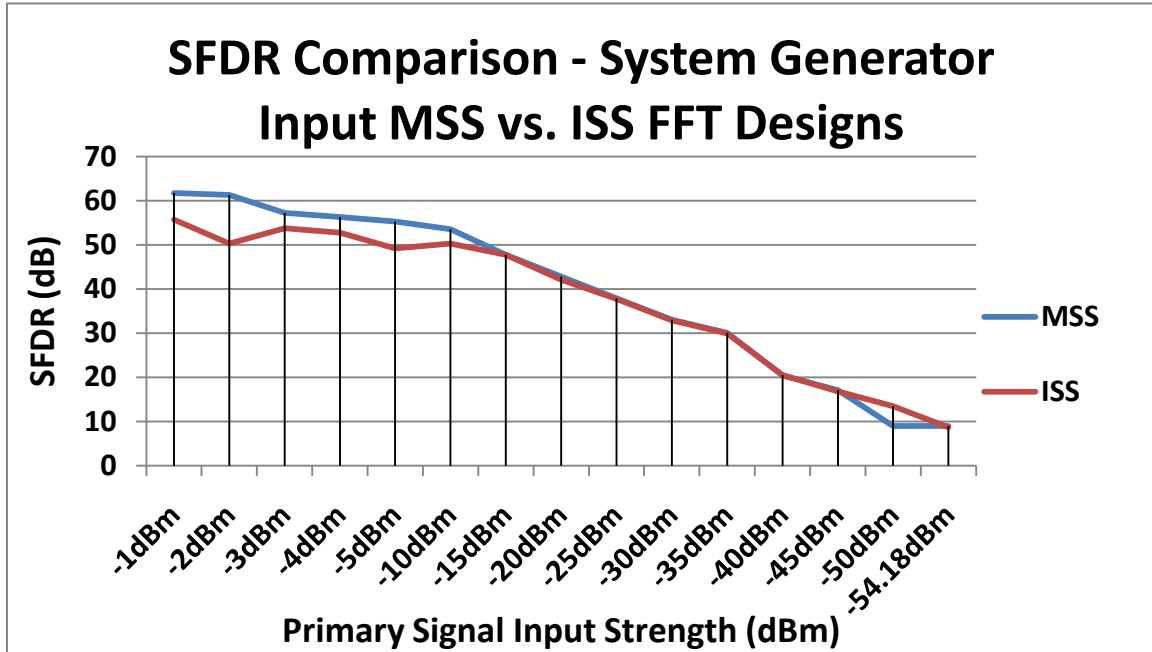


Figure 4.3 Plot showing performance of ISS and MSS receivers using ideal inputs

4.3 Performance Evaluation of Atmel Digitized Data for Multi-Stage Scaling and Single Stage Scaling Receiver Design

In order to more accurately determine the real world performance capabilities of the receiver, it was necessary to simulate both ISS and MSS receiver designs with non-ideal signal inputs. To accomplish this, digitized data was retrieved using our Atmel 10-bit ADC using inputs signals of varying magnitudes. The digitized data represents real-world figures that include noise and quantization errors that were not present in our Xilinx system generated ideal signals. Similar to the performance tests using the ideal signals, both ISS and MSS receiver designs were tested using our digitized data. This provided us with an accurate comparison for the performance capabilities of both designs. However, only digitized data composed of a single primary signal was available to use while simulating. Therefore, no evaluations for two-tone IDR performance for either receiver

design are presented in this thesis.

Single-tone digitized data was available and the performance of both receiver designs was charted for input signal strengths ranging from -7dBm down to -45dBm. For all signal strengths tested, the MSS receiver design matched or outperformed our ISS receiver design. Fig. 4.4 below plots the performance capabilities for both ISS and MSS receiver designs while using digitized 10-bit Atmel ADC data.

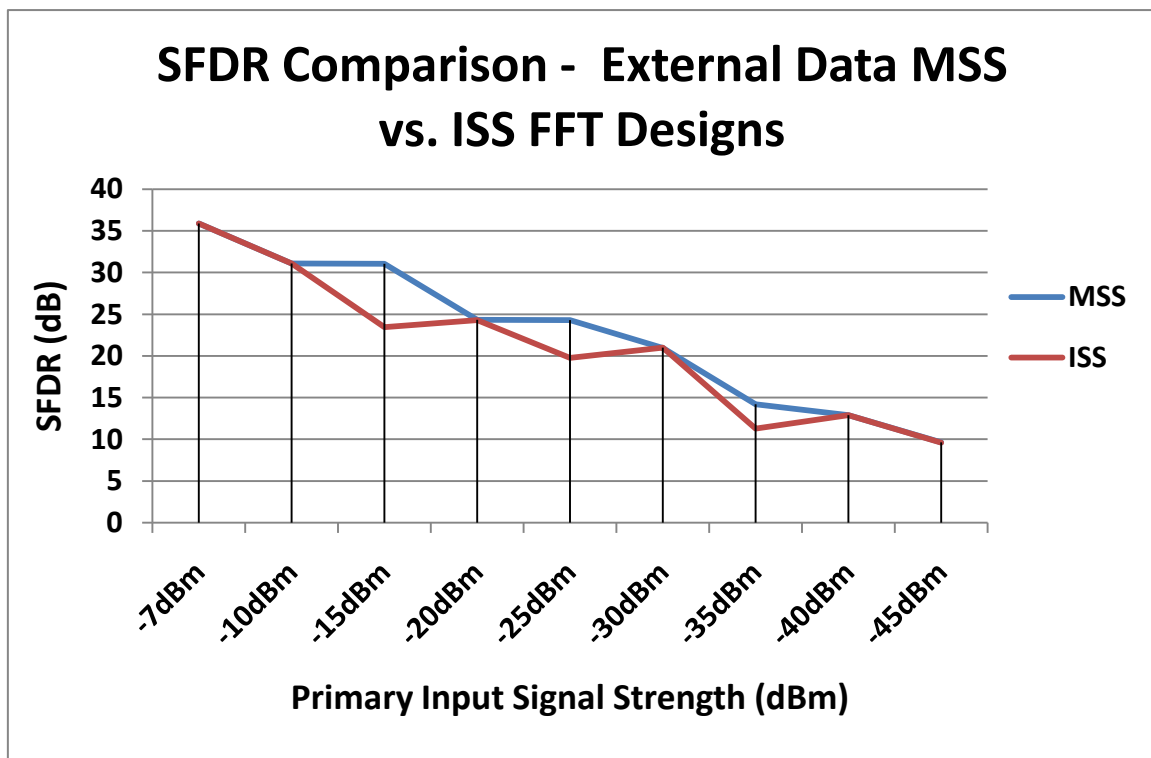


Figure 4.4 SFDR performance for ISS and MSS receiver designs using digitized ADC data

As shown by Figure 4.4, the MSS receiver design shows significant performance gains over the ISS receiver design for input signal strengths of -15dBm, -25dBm, and -35dBm. The maximum obtainable SFDR for both ISS and MSS receivers is 35.91dB for an input signal strength of -7dBm.

V. Design Flow and Prototyping Hardware

5.1 Design Flow

Design flow for the research started with a software based receiver design created within Matlab. Originally the receiver code was based on a previous receiver design that utilized a 128-point FFT, a 6-bit dynamic kernel, and permitted 8-bits of data from the ADC to travel within the stages of the FFT. The thresholding algorithm implemented in the original receiver code was based on a dual-thresholding methodology described previously in this thesis. Changes to the original receiver code were made to incorporate a 10-bit dynamic kernel and to allow 14-bits of data to pass between FFT stages. The implementation of our adaptive thresholding technique was also incorporated into our Matlab receiver code.

The second stage of design flow required the implementation of our receiver design created within Matlab to be ported into a Xilinx System Generator (XSG) design. XSG is a software extension of MATLAB's Simulink environment. XSG and Simulink contain a number of useful digital signal processing (DSP) blocks as well as logic gates and registers which provide a solid foundation for designing a microwave receiver.

This receiver design was completed in incremental steps. First an 8-point FFT was created utilizing a 10-bit dynamic kernel. The size of the FFT was continuously increased by powers of two, from an 8-point to 16-point FFT up to the final 128-point FFT version. Testing and verification for smaller designs was completed before porting them into a

larger FFT design. This streamlined the design process to aid in preventing design errors during the implementation phase. A scaling block was implemented after the final 128-point FFT had been completed and tested. Once the scaling block had been tested with the FFT, the thresholding block could be incorporated into the receiver design. This completed the design for the ISS receiver; multiple scaling blocks could then be added and verified to complete the design of the MSS receiver.

For the third stage of design flow, the creation of the very-high speed integrated circuit hardware design language (VHDL) used to represent our receiver design created in Xilinx System Generator was begun. This was accomplished with the help of Matlab and Xilinx System Generator. Xilinx System Generator is capable of creating VHDL code directly from the models and subsystems that make up the receiver design. This VHDL code can be used in conjunction with Xilinx ISE software to define a receiver model based in VHDL code to program our Virtex 4- SX55 FPGA board. A photo of the Virtex 4- SX55 FPGA board can be seen below in Fig. 5.1.



Figure 5.1 Photo of the Virtex 4- SX55 FPGA board [12]

Due to the large size of our receiver design, it needed to be broken into various subsystems in order for the Xilinx System Generator to be able to convert the design into VHDL. Our 128-point FFT MSS design was broken down into five subsystems for VHDL code generation. After successful VHDL code generation, the design was combined with our Xilinx Virtex 4 FPGA design kit, required to run our design and the Atmel 10-bit ADC together. After multiple trials, we were unable to successfully synthesize the 128-point MSS receiver design due to the computers not having sufficient quantities of system random access memory (RAM). Several attempts were also made to synthesize a 128-point ISS receiver design, however, the design was still too large to synthesize on our current machines. Due to system constraints beyond our control, our 128-point MSS and ISS designs were modified to incorporate a 64-point FFT to help reduce the size of the design. However, the 64-point MSS receiver design was still not able to synthesize successfully on any of our computer systems. The 64-point ISS design was synthesized successfully, and a detailed report of the overall hardware usage can be seen on the next page in Fig. 5.2.

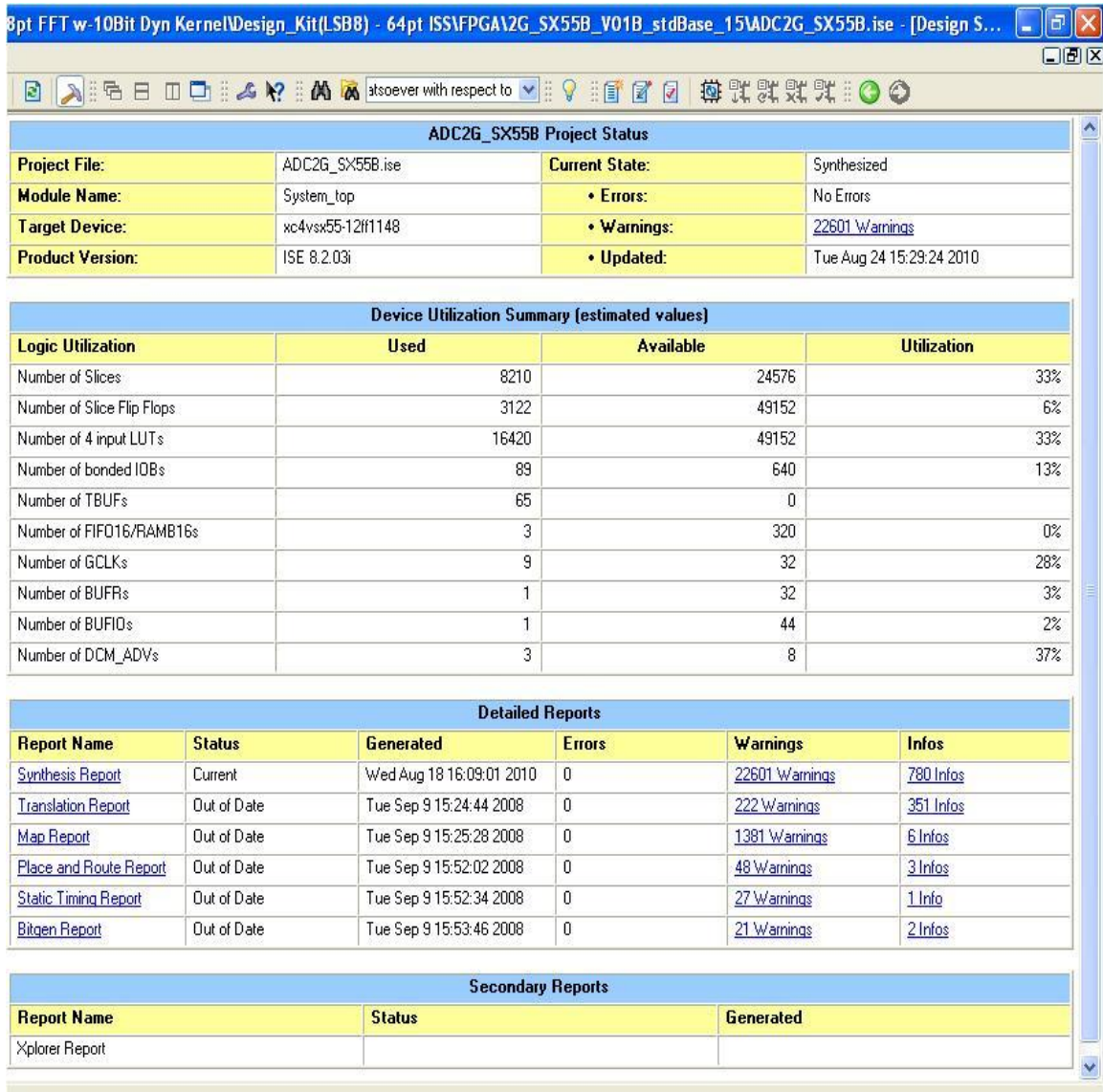


Figure 5.2 Xilinx ISE synthesis hardware report for 64-point ISS receiver design

As shown in Figure 5.2, overall hardware requirements for the 64-point ISS receiver design on a Virtex 4 - SX55 are roughly at 33%. Even though we could not successfully synthesize larger designs utilizing a 128-point FFT, these results show good promise that our original 128-point FFT designs should fit on the Virtex 4 FPGA.

The fourth step in our design flow involves mapping and completing a timing verification of the receiver design. All steps prior to this one have already been successfully completed. If the timing verification fails, it will be necessary to add pipelines within the receiver design so that it can meet our timing specifications. If the design is not successfully mapped, it will require us to minimize the hardware usage of the design so that it can be successfully mapped and routed on the FPGA. After any necessary changes have been made to the design it will then be possible to create a bit stream file within ISE to program our Virtex 4 FPGA board with the receiver design.

The fifth and final process involves the testing and verification of the receiver design once it has been programmed onto our Virtex 4 FPGA board. This verification can be accomplished with the use of Xilinx Chipscope debugging cores that are implemented within the VHDL code of the receiver. After the FPGA board has been properly programmed, it will then be possible to use Xilinx's Chipscope Analyzer to verify that our design is working properly.

5.2 Prototyping Hardware

The target platform for prototyping our designs is the Delphi ADC3255 PCI Mezzanine card (PMC). This board contains a combination of a 10-bit Atmel ADC capable of sampling at 2.56GHz and a Virtex 4 - SX55FPGA board. A top level view of the Delphi ADC3255 board can be seen in Fig. 5.3 located on the following page.

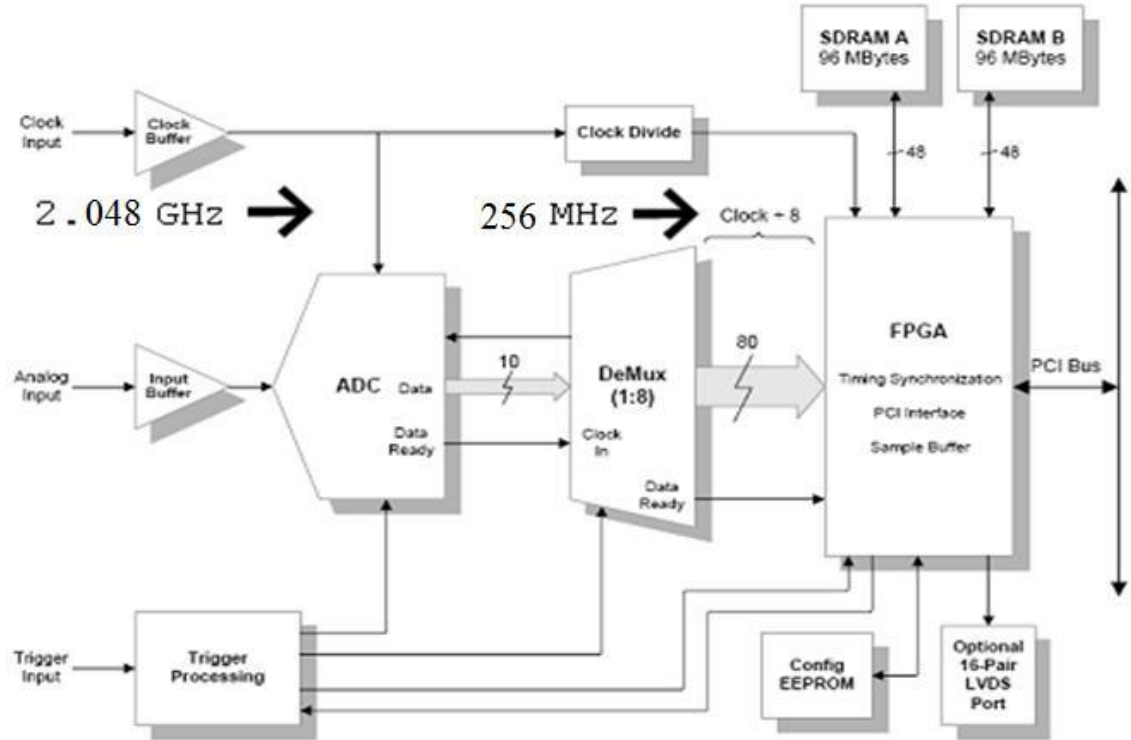


Figure 5.3 Top level view of a Delphi ADC3255 board [11]

For this research, the Atmel ADC will be set to sample at 2.048 GHz. As shown from Figure 5.3, the digital data is first sent through a 1:8 demultiplexer block which will divide the 2.048GHz clock by 8, producing an operating frequency of 256MHz. An external clock generator will be used to source the 2.048GHz clock frequency required for the board.

VI. Conclusion

6.1 Contribution

The research contributed forth from this thesis has provided two microwave radar receiver models based in a Xilinx System Generator platform. These designs are considered wide-band as their bandwidth exceeds 1 GHz in the radio frequency (RF) spectrum. With the incorporation of a 128-point FFT that utilizes a 10-bit dynamic kernel function and allows 14-bits of data to pass between FFT stages, significant receiver performance improvements have been achieved when compared to previous receiver designs. The mainstay of this design has been focused around the implementation and optimization of an adaptive thresholding algorithm capable of operating in a non-ideal environment. This thesis provides two microwave receiver designs that incorporate slightly varying adaptive thresholding techniques. The successful simulations of these designs using digitized data from our target platform 10-bit Atmel ADC shows our MSS and ISS receiver design are capable of achieving an SFDR of 35.91dB for a primary signal strength of -5dBm. Research has also shown that these designs are capable of obtaining a mono-tone sensitivity of -45dBm while maintaining a near 10dB SFDR. Synthesis of our 64-point ISS microwave receiver design shows promising results, with hardware usage on our Virtex 4 - SX55 at roughly 33%. MSS receiver design has also shown nearly an improvement for all signal strengths ranging from -7dBm down to -45dBm with minimal increase in overall hardware usage.

6.2 Future Work

Future work on our microwave receiver design will continue as I enter the PhD program for engineering at Wright State University. Synthesis, along with timing analysis and design mapping will be completed to allow for testing of the design once it has been loaded on the Virtex 4 - SX55 board. Simulations will be completed to show receiver performance with regards to it's two-tone IDR. Future contributions will also include the study of mixed CW and pulsed waves (PW) and their effects on overall receiver performance. The addition of multi-tone signal performance beyond the dual-tone signals performance evaluations presented in this paper will also be studied.

Future enhancements to current receiver designs may include the introduction of a variable truncation scheme (VTS) between FFT stages to allow for greater data retention and better receiver performance. VTS will also allow the portability of lower input bit-width receivers within our design. Further improvements may include continued optimization on the number of scaling stages present, to further decrease hardware requirements.

References

- [1] J. B. Y. Tsui and J. P. Stephens, "Digital microwave receiver technology," IEEE Trans. on Microwave Theory and Techniques, vol. 50, no. 3, pp. 699-705, Mar. 2002.
- [2] C.-I. H. Chen, K. George, W. McCormick, J. B. Y. Tsui, S. L. Hary, and K. M. Graves, "Design and performance evaluation of a 2.5-GSPS digital receiver," IEEE Trans. on Instrumentation and Measurement, vol. 54, no. 3, pp. 1089-1099, Jun. 2005.
- [3] K. George, C.-I. H. Chen, and J. B. Y. Tsui, "Extension of two-signal spurious-free dynamic range of wideband digital receivers using Kaiser window and compensation methods," IEEE Trans. On Microwave Theory and Techniques, vol. 55, no. 4, pp. 788-794, Apr. 2007.
- [4] M. A. Sanchez, M. Garrido, M. Lopez-Vallejo, J. Grajal, and C. Lopez-Barrio, "Digital channelised receivers on FPGA platforms," in Proc. of IEEE International Radar Conference, pp. 816-821, May 2005.
- [5] A. Farina and F. A. Studer, "A review of CFAR detection techniques in radar systems," Microwave Journal, vol. 29, no. 9, pp. 115-128, Sept. 1986.
- [6] A. Farina and L. Ortenzi, "Effect of ADC and Receiver Saturation on Adaptive Spatial Filtering of Directional Interference," Signal Processing, vol. 83, no. 5, pp. 1065-1078, May 2003.
- [7] A. De Maio, G. Foglia, E. Conte, and A. Farina, "CFAR behavior of adaptive detectors: an experimental analysis," IEEE Trans. On Aerospace and Electronic Systems, vol. 41, issue 1, pp. 233-251, Jan. 2005.
- [8] Y.-H. G. Lee and C.-I. H. Chen, "Dual Thresholding for Digital Wideband Receivers with Variable Truncation Scheme", Proceedings of IEEE International Symposium on Circuits and Systems, pp. 920-923, Taipei, Taiwan, May 2009.
- [9] S. Benson, S. and C.-I. H. Chen, "Adaptive Thresholding for High Dual-Tone Signal Instantaneous Dynamic Range in Digital Wideband Receiver," Proceedings of IEEE International Instrumentation and Measurement Technology Conference, pp. 616-619, Austin, Texas, May 2010
- [10] Xilinx Virtex-4 FPGA Overview. Online,
http://www.xilinx.com/support/documentation/data_sheets/ds112.pdf
- [11] Delphi Engineering Group, Inc., ADC3255 User Manual, v1.10, 2006.
- [12] Delphi Engineering Group, Inc., Online
www.delphieng.com/
- [13] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of the complex Fourier series," Math. Of Computation, vol. 19, no. 90, pp. 297-301, Apr. 1965.

- [14] Despain, Alvin M. "Very Fast Fourier Transform Algorithms Hardware for Implementation," IEEE Trans. On Computers, 28(5):333-341.1979.
- [15] Oppenheim, A. V., R. W. Schafer. Discrete-Time Signal Processing, (First Edition), Englewood Cliffs, NJ: Prentice Hall, Inc., 1989.
- [16] D. Pok, C.-I. H. Chen, J. Schamus, C. Montgomery and J. B. Y. Tsui, "Chip Design for Monobit Receiver," IEEE Trans. Microwave Theory and Techniques, vol. 45, no. 12, pp. 2283-2295, December 1997.
- [17] Ryan Bone. "FPGA design of a hardware efficient pipelined FFT processor." M.S. thesis, Wright State University, U.S.A., 2008.
- [18] Lee YuHeng(George Lee). "Dynamic Kernel Function for High-Speed Real-Time Fast Fourier Transform." Ph.D dissertation, Wright State University, U.S.A., 2009.